OIPE JC189
MAY 29 2001
PATENT & TRADEMARK OFFICE

No new matter has been added by the foregoing amendment. The amendments were made to correct inadvertent typographical errors in the specification and to connect the tables following paragraph [00241] to Fig. 14.

Applicants respectfully request that the above-noted amendments be entered.

Respectfully submitted,

Dated: 5/23/2001

By: James K Weixel
James K. Weixel, Reg. No. 44,399

VERIZON SERVICES GROUP
600 Hidden Ridge, HQE03H01
Irving, TX 75038
781-466-2220

EXHIBIT 1 - VERSION WITH MARKINGS TO SHOW CHANGES MADE TO SPECIFICATION

[00149] Processing then moves to step 152 which initiates a loop for every nesting relationship returned in the recordset selected in step 148. Processing then moves to decision block 154 which determines whether the ratio of elements in the particular nesting relationship is a 1-to-1 or a ~~1 to n~~ 1-to-n relationship. If the decision block 154 determines that the ratio is 1-to-1, processing moves to step 156 in which a foreign key is inserted in the parent table of the relationship. A proposed SQL statement to accomplish this task is shown in note 158 associated with step 156 in Fig. 8. If the decision block 154 determines that the ratio is ~~1 to n~~ 1-to-n, processing moves to step 160 in which a foreign key is inserted into the child table in the relationship. A proposed SQL statement to accomplish this task is shown in note 162 associated with step 160 in Fig. 8. In either case, processing then moves to re-connector 164 and then to decision block 166 which determines whether additional nesting relationships need to be processed. If so, processing returns to step 152. If not, processing ends.

[00183] The following data dictionary representative of the table schema 22 is thereby generated in accordance with this inventive method:

Table Name	Required Columns	Data Columns	Relationship Columns
PCDATA	iid, order	value	
book	iid, order	booktitle	G1_iid
booktitle	iid, order	PCDATA_value	
article	iid, order	title	contactauthor_iid
title	iid, order	PCDATA~~.~~_value	
contactauthor	iid, order		
monograph	iid, order	title	author_iid, editor_iid
editor	iid, order	name	
author	iid, order	id, name_firstname, name_lastname	parent_G1_iid
name	iid, order	firstname, lastname	
firstname	iid, order	PCDATA_value	
lastname	iid, order	PCDATA_value	
affiliation	iid, order		
G1	iid		editor_iid

G2	<u>iid</u>		parent_article_iid, author_iid, affiliation_iid
G3	<u>iid</u>		parent-editor_iid, book_iid, monograph_iid
AG	<u>iid</u>	PCTDATA_value, booktitle, title, firstname, lastname, name_firstname, name-_lastname	parent_affiliation_iid, book_iid, article_iid, contactauthors_iid, monograph_iid, editor_iid, author_iid, affiliation_iid
contactauthors .authorIDs	<u>idd</u>	value	PARENT_contactauthors_idd

[00200] The following tables provide the pattern-mapping table 36 with all of the patterns from the metadata tables 34 generated according to the DTD 18 provided in Example 1 and discussed throughout. By way of clarification, different types of patterns have been separated by a solid line in the following table in the following order: (1) node patterns; (2) attribute patterns; and (3) link patterns.

Pattern
PCDATA
book
booktitle
article
title
contactauthors
monograph
editor
author
name
firstname
lastname
affiliation
contactauthors.authorIDs
editor.name
author.id
PCDATA.value
book→booktitle
book→author—
book→editor—
booktitle→PCDATA
article→title
article→author
article→affiliation
article→contactauthors

```
title→PCDATA
monograph→title
monograph→author
monograph→editor
editor→book
editor→monograph
author→name
name→firstname
name→lastname
firstname→PCDATA
lastname→PCDATA
affiliation→PCDATA
affiliation→book
affiliation→booktitle
affiliation→article
affiliation→title
affiliation→contactauthors
affiliation→monograph
affiliation→editor
affiliation→author
affiliation→name
affiliation→firstname
affiliation→lastname
affiliation→affiliation
```

[00209] When a link is encountered, three possible cases result. First, the foreign key in one table can be updated with the key value in another table. Second, if there is a group in this link, then a new tuple is created in the group table as well as the corresponding foreign keys are updated. Third, if the child node is inlined in the parent node, then all of the attributes of the child table are copied into the parent table. The details of how to generate those actions are discussed below.

Pattern	Actions
Node: T	create(T)
Attribute:T.A	update(T.A) +|_decompose decompose(T.value), assign(T_A.parent.T_iid, T.iid))
Link: A→B	assign(A.iid, B.iid) +|_(create(G), assign(A.G_iid, G.iid), assign(G.B_iid, B.iid) +|_(create(G), assign(A.G_iid, G.iid), assign(B.parent_G_iid, G.iid) +|_(create(G), assign(G.parent_A_iid, A.iid), assign(G.B_iid, B.iid)

	+⌊ (create(G), assign(G.parent_A_iid, A.iid), assign(B.parent_G_iid, G.iid) +⌊ assign(A.attribute, B.attribute)

[00222] By way of summary, the pattern mapping tables 36 are initialized by putting the generated pattern and the corresponding actions together. For all of the node patterns, e.g., `T, put action create(T)`. For all of the attribute patterns, e.g., `T.A, put action update(T.A)`. For the link pattern, e.g., A→B, there are different cases. If the link pattern is not related to a group, then based on the mapping rule, if the relationship is one-to-one, then `put assign(A.B_iid, B.iid)`, if the relationship is one-to-many, then `put assign(B.parent-_A_iid, A.iid)`. If the link pattern is related to a group G, then put G first, then handle the relationship between A→G and G→B separately as described before.

[00228] The pattern mapping table 36 generated from the metadata described in the example follows:

Pattern	Actions
PCDATA	create(PCDATA)
book	create(book)
booktitle	create(booktitle)
article	create(article)
title	create(title)
contactauthors	create(contactauthors)
monograph	create(monograph)
editor	create(editor)
author	create(author)
name	create(name)
firstname	create(firstname)
lastname	create(lastname)
affiliation	create(affiliation)
contactauthors.authorIDS	update(contactauthor.authorIDs)
editor.name	update(editor.name)
author.id	update(author.id)
PCDATA.value	update(PCDATA.value)
book→booktitle	assign(book.booktitle_iid, booktitle.iid)
book→author—	create(G1), assign(book.G1_iid, G1.iid), assign(author.parent_G1_iid = G1.iid)
book→editor—	create(G1), assign(book.G1_iid, G1.iid), assign(G1.editor.iid, editor.iid)
booktitle→PCDATA	assign(booktitle.PCDATA_iid, PCDATA.iid)
article→title	assign(article.title_iid, title.iid)
article→author	create(G2), assign(G2.parent_article_iid, article.iid), assign(G2.author_iid, author.iid)

article→affiliation	create(G2), assign(G2.parent_article_iid, article.iid), assign(G2.affiliation_iid, affiliation.iid)
article→contactauthors	assign(article.contactauthors_iid, contactauthors.iid)
title→PCDATA	assign(title.PCDATA_iid, PCDATA.iid)
monograph→title	assign(monograph.title_iid, title.iid)
monograph→author	assign(monograph.author_iid, author.iid)
monograph→editor	assign(monograph.editor_iid, editor.iid)
editor→book	create(G3), assign(G3.parent_editor_iid, editor.iid), assign(G3.book_iid, book.iid)
editor→monograph	create(G3), assign(G3.parent_editor_iid, editor.iid) assign(G3.monograph_iid, monograph.iid)
author→name	assign(author.name_iid, name.iid)
name→firstname	assign(name.firstname_iid, firstname.iid)
name→lastname	assign(name.lastname_iid, lastname.iid)
firstname→PCDATA	assign(firstname.PCDATA_iid, PCDATA.iid)
lastname→PCDATA	assign(lastname.PCDATA_iid, PCDATA.iid)
affiliation→PCDATA	create(AG), assign(A.G.parent_affiliation_iid, affiliation.iid), assign(AG.PCDATA_iid, PCDATA.iid)
affiliation→book	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.book_iid, book.iid)
affiliation→booktitle	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.booktitle_iid, booktitle.iid)
affiliation→article	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.article_iid, article.iid)
affiliation→title	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.title_iid, title.iid)
affiliation→ contactauthors	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.contactauthors_iid, contactauthors.iid)
affiliation→monograph	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.monograph_iid, monograph.iid)
affiliation→editor	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.editor_iid, editor.iid)
affiliation→author	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.author_iid, author.iid)
affiliation→name	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.name_iid, name.iid)
affiliation→firstname	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.firstname_iid, firstname.iid)
affiliation→lastname	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.lastname_iid, lastname.iid)
affiliation→affiliation	create(AG), assign(AG.parent_affiliation_iid, affiliation.iid), assign(AG.affiliation_iid, affiliation.iid)

[00229] Then, the following pattern-mapping table results:

Pattern	Actions
PCDATA	create(PCDATA)
book	create(book)
booktitle	create(booktitle)
article	create(article)
title	create(title)
contactauthors	create(contactauthors)
monograph	create(monograph)
editor	create(editor)
author	create(author)
name	create(name)
firstname	create(firstname)
lastname	create(lastname)
affiliation	create(affiliation)
contactauthors.authorIDS	decompose(contactauthors_authorIDs.value), assign(contactauthors_authorIDs.parent_. contactauthors_iid, contactauthors.iid)
editor.name	update(editor.name)
author.id	update(author.id)
PCDATA.value	update(PCDATA.value)
book→booktitle	assign(book.booktitle_iid, booktitle.iid)
book→author—	create(G1), assign(book.G1_iid, G1.iid), assign(author.parent_G1_iid = G1.iid)
book→editor—	create(G1), assign(book.G1_iid, G1.iid), assign(G1.editor.iid, editor.iid)
booktitle→PCDATA	assign(booktitle.PCDATA_value, PCDATA.value)
article→title	assign(article.title, title.PDATA_value)
article→author	create(G2), assign(G2.parent_article_iid, article.iid), assign(G2.author_iid, author.iid)
article→affiliation	create(G2), assign(G2.parent_article_iid, article.iid), assign(G2.affiliation_iid, affiliation.iid)
article— →contactauthors	assign(article.contactauthors_iid, contactauthors.iid)
title→PCDATA	assign(title.PCDATA_value, PCDATA.value)
monograph→title	assign(monograph.title, title.PCDATA_value)
monograph→author	assign(monograph.author_iid, author.iid)
monograph→editor	assign(monograph.editor_iid, editor.iid)
editor→book	create(G3), assign(G3.parent_editor_iid, editor.iid), assign(G3.book_iid, book.iid)
editor→monograph	create(G3), assign(G3.parent_editor_iid, editor.iid) assign(G3.monograph_iid, monograph.iid)
author→name	assign(author.name_firstname, name.firstname), assign(author.name_lastname, name.lastname)
name→firstname	assign(name.firstname, firstname.PCDATA_value)
name→lastname	assign(name.lastname, lastname.PCDATA_value)
firstname→PCDATA	assign(firstname.PCDATA_value, PCDATA.value)
lastname→PCDATA	assign(lastname.PCDATA_value, PCDATA.value)
affiliation→PCDATA	create(AG), assign(A.G.parent_affiliation_iid,

	affiliation.iid), assign(AG.PCDATA_iid, PCDATA.iid)
affiliation→book	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.book_iid, book.iid)
affiliation→booktitle	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.booktitle_iid, booktitle.iid)
affiliation→article	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.article_iid, article.iid)
affiliation→title	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.title_iid, title.iid)
affiliation→ contactauthors	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.contactauthors_iid, contactauthors.iid)
affiliation→monograph	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.monograph_iid, monograph.iid)
affiliation→editor	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.editor_iid, editor.iid)
affiliation→author	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.author_iid, author.iid)
affiliation→name	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.name_firstname, name.firstname), assign (AG.name_lastname, name.lastname)
affiliation→firstname	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.firstname, firstname.PCDATA-value)
affiliation→lastname	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.lastname, lastname.PCDATA_value)
affiliation→affiliation	create(AG), assign(AG.parent affiliation iid, affiliation.iid), assign(AG.affiliation_iid, affiliation.iid)

[00241] It should be noted that, because the elements of the data are the basic units in the XML document 12, the system 10 should still store the data of the corresponding elements into their tables during the loading process. However, in the case of inline attributes, if the element is in-lined into an attribute, then the table of that element is no longer used after the loading. Therefore, those unused tables could be deleted after loading the data. The following table shows the result of the data loading in the relational tables:

~~book~~

~~iid~~	~~Order~~	~~booktitle~~	~~GI_iid~~
~~1~~	~~33~~	~~the XML Handbook~~	~~1~~

article

iid	Order	title	contactauthor.idd
1	1	XML Relation Mapping	1

~~editor~~

~~iid~~	~~Order~~	~~name~~
~~1~~	~~32~~	~~Patti Guerrieri~~

contactauthors

iid	Order
1	~~48~~28

~~G3~~

~~iid~~	~~parent_editor_idd~~	~~book.iid~~	~~monograph_idd~~
~~1~~	~~1~~	~~1~~	

~~Monograph~~

~~iid~~	~~order~~	~~title~~	~~author.iid~~	~~editor_idd~~
~~1~~	~~23~~	~~Repository Support for Metadata-based Legacy Migration~~	~~4~~	~~1~~

author

iid	order	id	name_firstname	name_lastname	parent_GI_idd
1	4	xz	Xin	Zhang	
2	~~10~~12	gm	Gail	Mitchell	
3	~~16~~20	wl	Wang-chien	Lee	
~~4~~	~~26~~	~~sh~~	~~Sandra~~	~~Heiler~~	
~~5~~	~~36~~	~~cg~~	~~Charles F.~~	~~Goldfarb~~	~~1*~~
~~6~~	~~42~~	~~pp~~	~~Paul~~	~~Prescod~~	~~1*~~

~~*This field is filled with the techniques of handling multi-level grouping that is not addressed in this application.~~

G2

iid	parent_article.iid	author.idd	affiliation.idd
1	1	1	1
2	1	2	2
3	1	3	3
~~4~~	~~1~~		~~1~~

contactauthors.authorIDS

iid	Value	Parent_contactauthors_idd
1	xz	1
2	~~g~~gm	1
3	~~ml~~wl	1

affiliation

iid	Order
1	~~22~~10
2	18
3	26

~~G1~~

~~iid~~	~~editor.idd~~
~~1~~	

AG

idd	PCDATA. value	book-title	title	first-name	last-name	name.first-name
1	Department ... 2280					
2	Verizon ... 02451					
3	Verizon ... 02451					

AG (continued)

name.last-name	parent-_~~related~~ -~~work~~ affilia-tion.idd	book.idd.idd	contact-authors.idd	mono-graph.idd	editor.idd	author.idd
	1 2 3			~~1~~		

EXHIBIT 3 - MARKED-UP COPY OF AMENDED DRAWINGS

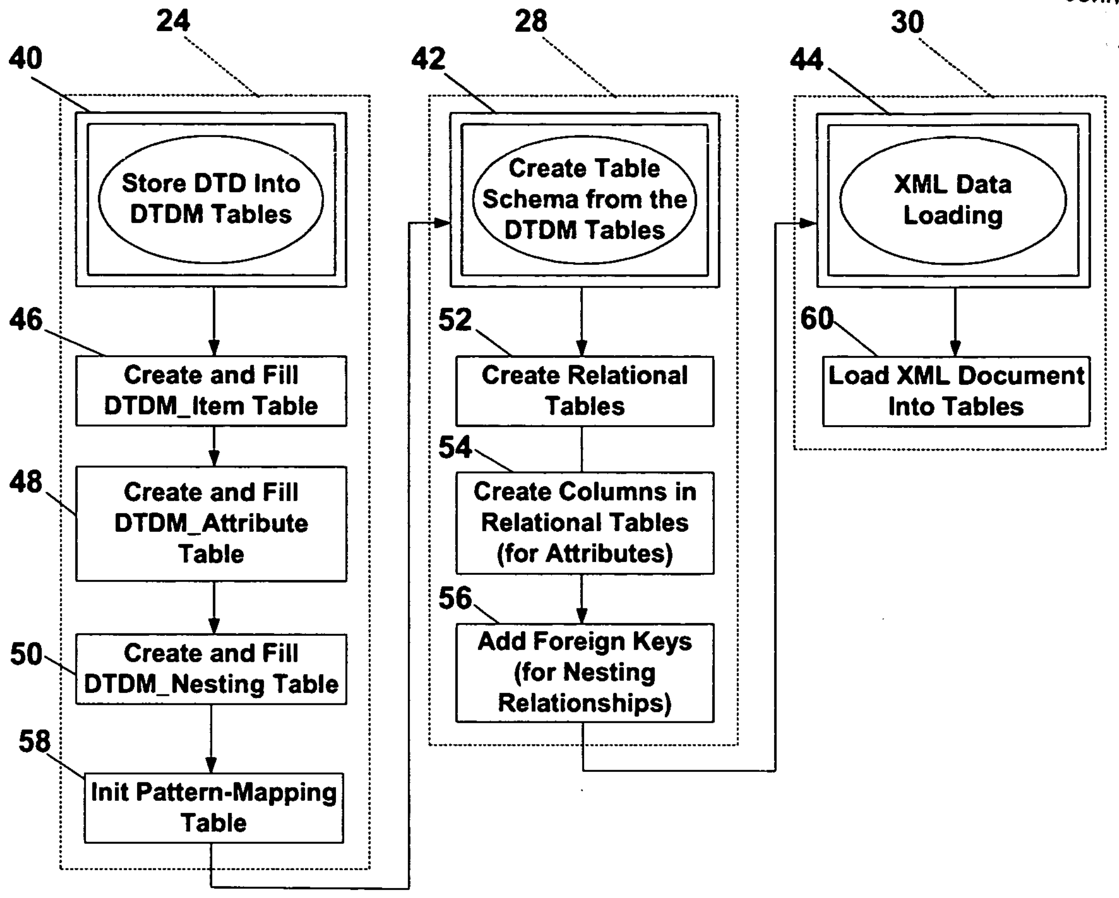
24
28
30
40
42
44
Store DTD Into DTDM Tables
Create Table Schema from the DTDM Tables
XML Data Loading
46
Create and Fill DTDM_Item Table
48
Create and Fill DTDM_Attribute Table
50
Create and Fill DTDM_Nesting Table
58
Init Pattern-Mapping Table
52
Create Relational Tables
54
Create Columns in Relational Tables (for Attributes)
56
Add Foreign Keys (for Nesting Relationships)
60
Load XML Document Into Tables

Fig. 2

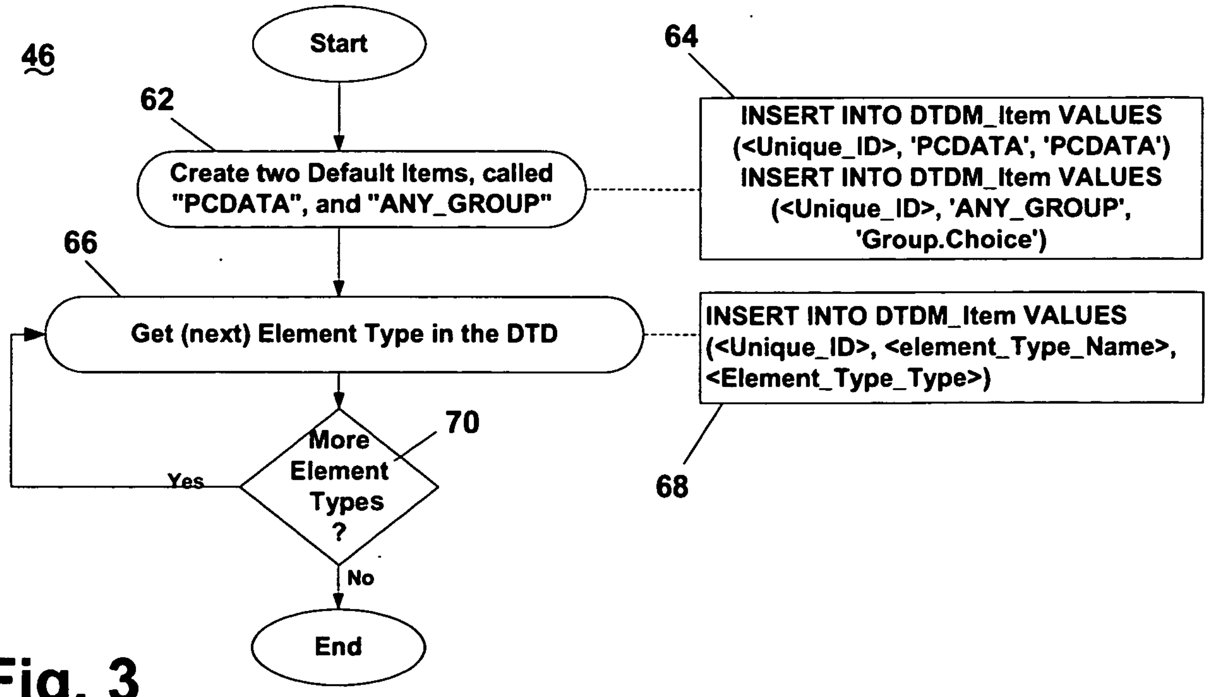
46
Start
62
Create two Default Items, called "PCDATA", and "ANY_GROUP"
64
INSERT INTO DTDM_Item VALUES (<Unique_ID>, 'PCDATA', 'PCDATA') INSERT INTO DTDM_Item VALUES (<Unique_ID>, 'ANY_GROUP', 'Group.Choice')
66
Get (next) Element Type in the DTD
INSERT INTO DTDM_Item VALUES (<Unique_ID>, <element_Type_Name>, <Element_Type_Type>)
68
70
More Element Types ?
Yes
No
End

Fig. 3

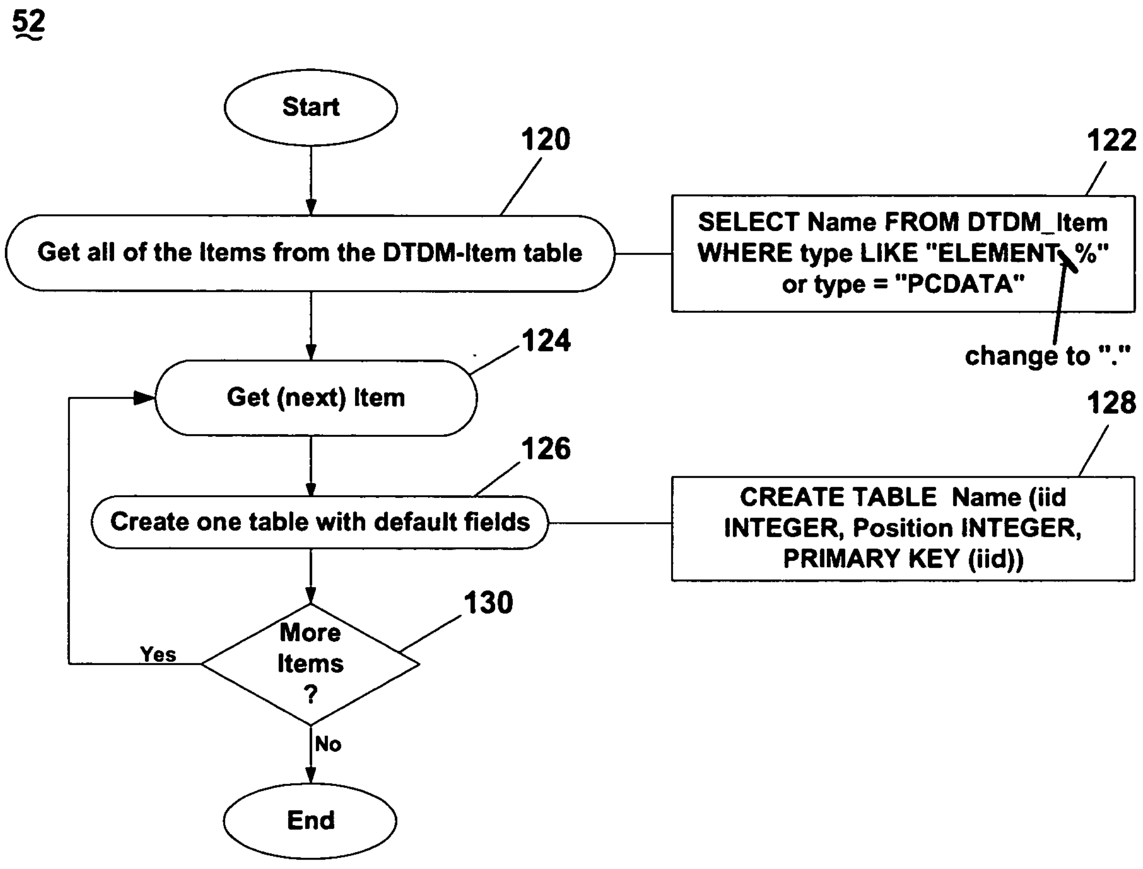
52
Start
120
122
Get all of the Items from the DTDM-Item table
SELECT Name FROM DTDM_Item WHERE type LIKE "ELEMENT_%" or type = "PCDATA"
change to "."
124
Get (next) Item
126
128
Create one table with default fields
CREATE TABLE Name (iid INTEGER, Position INTEGER, PRIMARY KEY (iid))
130
More Items ?
Yes
No
End

Fig. 6

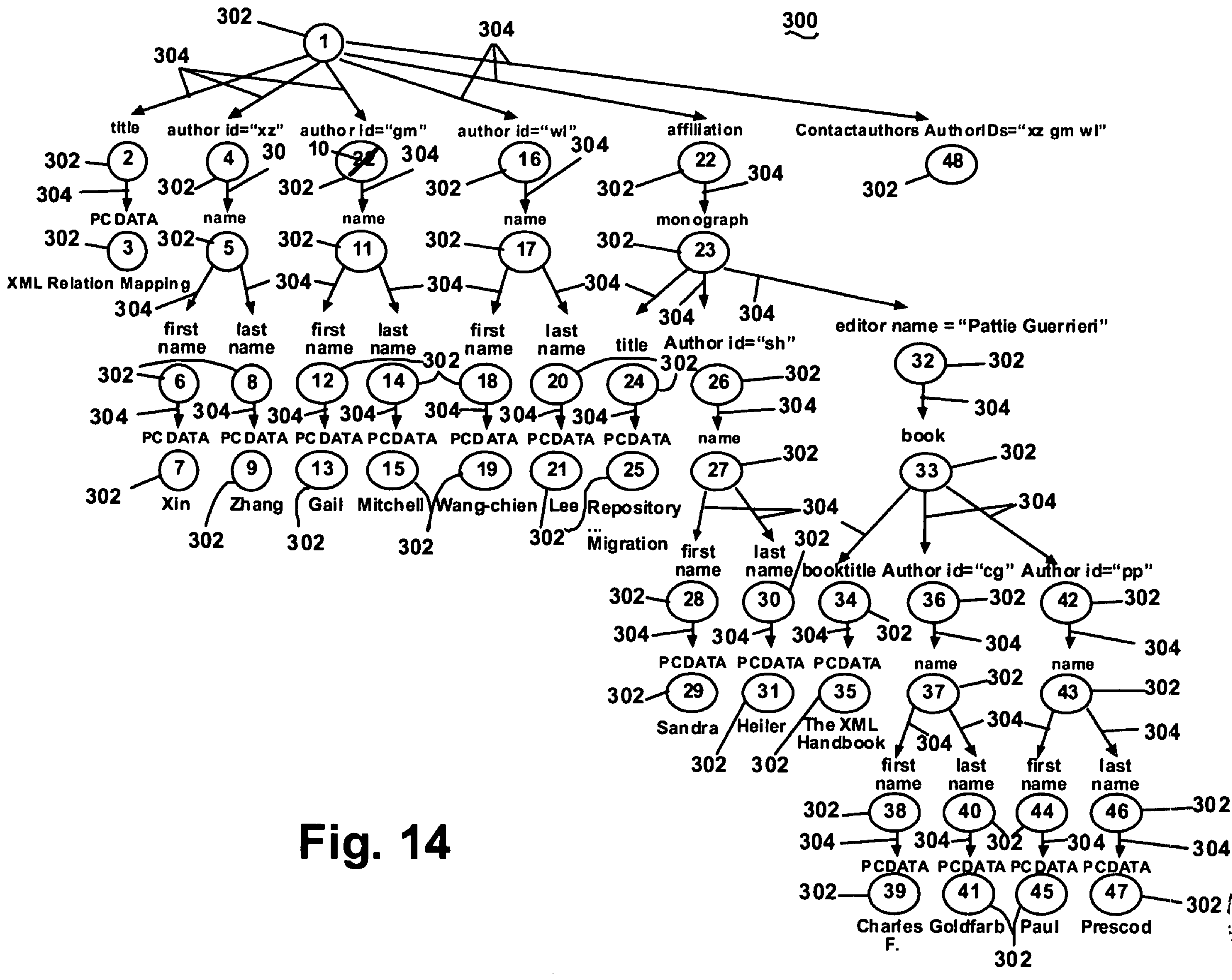
300
1
title
author id="xz"
author id="gm"
author id="wl"
affiliation
Contactauthors AuthorIDs="xz gm wl"
PCDATA
XML Relation Mapping
name
monograph
first name
last name
title
Author id="sh"
editor name = "Pattie Guerrieri"
book
PCDATA
Xin
Zhang
Gail
Mitchell
Wang-chien
Lee
Repository … Migration
booktitle
Author id="cg"
Author id="pp"
Sandra
Heiler
The XML Handbook
Charles F.
Goldfarb
Paul
Prescod
302
304
10
30

Fig. 14